

xAPI Profile Server Authoring Guide

Version 2.0

27 October 2020

Under contract #HQ0034-19-C-0122 by



xAPI Profile Server Authoring Guide

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, this is distributed under the License and is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Overview	3
Methodology	5
Prioritizing What To Measure	5
Recognizing Likely Reasons for a new xAPI Profile	7
Authoring Guidance	8
Authoring an xAPI Profile in “Easy” Steps	8
Implementation Notes	9
Statement Templates	10
Determining Properties	11
Rules	12
Concepts	14
Patterns	16

Overview

About the xAPI Profile Server Authoring Guide

This document provides the reader context for the use of the xAPI Profile Server as an authoring support for developing xAPI Profiles. In addition to providing services to validate that a .json-ld xAPI Profile is conforming to the xAPI Profiles specification and generally making hosted xAPI Profiles and lookup available, the xAPI Profile Server is an authoring tool for profiles, guiding users with best practices designed into the workflow to help authors produce strong xAPI Profiles.

This document is written with the following assumptions about its reader:

- The reader has a ready and working knowledge of the xAPI specification.
- The reader has a ready and working knowledge of the xAPI Profiles Specification
- The reader has a ready and working knowledge of why and how one authors an xAPI Profile.

While this document does not propose a specific pedagogy, methodology or ideological framing for the development of an xAPI Profile, it does provide context, the authors' principles made explicit, and practical implementation considerations for authoring xAPI Profiles given that framing.

The xAPI Profile Server Library

With the release of the xAPI Profile Server, a number of documents in the xAPI Profile Server Library will be helpful in different ways.

- The **xAPI Profile Server User Guide** provides a comprehensive set of step-by-step walk-throughs detailing end-user features of the xAPI Profile Server.
- The **xAPI Profile Server Administrator Guide** provides focused step-by-step walk-throughs for managing an xAPI Profile Server instance.
- This document, the **xAPI Profile Server Authoring Guide**, documents initial recommended practices in using the xAPI Profile Server to develop xAPI Profiles.

How to Read This Document

The xAPI Profile Server Authoring Guide is organized by areas of anticipated needs for xAPI Profile Authors. **Methodology** is important to establish so deliberate, transparent decisions can be accounted for in periodic evaluation of the program's success, and **Authoring Guidance** that aims to reduce, if not eliminate, technical and semantic interoperability challenges in the near and long term.

This document places information referenced within the **xAPI Profile Server User Guide** into more practical guidance to author xAPI Profiles. At the risk of competing information sources, any step-by-step instructions are described in the xAPI Profile Server User Guide. This document references such instructions directly to the page and heading of accompanying documents.

Where distinctions in the aims of authoring an xAPI Profile demonstrate considerations for exceptions, this document will attempt to illustrate the differences and, where it matters, offer a preference for what is considered best practice at the time of publication.

Methodology

As xAPI Profiles, to some extent, codify what an organization values in human performance, when one implements an xAPI Profile, one implements a translucent set of rules that bind a learning experience. Ideally these should reflect the values of the organization responsible for producing the xAPI Profile. These rules quickly become invisible to the learner and even the stakeholders themselves. These rules quickly shape operating behaviors. Any bias or logical flaw will surface as reporting, and decisions based on such reporting, play out over time. An xAPI Profile can impact how learning and work performance are metrically evaluated by leaders. This completely removes from the visibility or perspective of the people doing the work being evaluated. As a result, it is an ethical, financial, and potentially liability imperative to be transparent and to communicate well in the process of defining xAPI Profiles in order to provide meaningful metrics for evaluation.

Demonstrably, this section of this particular document in the xAPI Profile Server Library attempts to expose such tacit and implicit information for the reader to provide an architect's perspective. This is done in hope of helping the reader to understand the difference between what is technically capable and what is pragmatically sound, so future technology and practice may mature independently.

Prioritizing What To Measure

The authoring of xAPI Profiles *should* mature into a skilled professional practice, with refined pedagogical models that stakeholders, practitioners and even learners alike may broadly recognize and understand.

This document does not describe a process and methodology to gather the research needed to inform an xAPI Profile. At the time of this writing, however, xAPI Profiles are in their infancy and while this document does not illustrate a specific methodology in the development of an xAPI Profile, it does provide context, principles and practical implementation considerations for authoring xAPI Profiles. The pioneers working with xAPI Profiles since 2017 suggest that it is not good practice to track what's easy to identify or simply track what engineers in a specification group suggest. Instead the author should answer the question "What should we measure?" This is done by considering the value the organization hopes to realize for itself, if not for the learners themselves.

Analytics, in general, reflects how an organization demonstrates accountability. To that end the effectiveness of an xAPI Profile is largely dependent on high correlation between what people in the organization say they value in research, and how those people assign value in practice. In other words, it's important to capture requirements that translate to solutions with measurable success criteria across observable developmental milestones.

There are three vectors to consider what to measure with analytics. There's a *financial* consideration which in practice is likely to be a gatekeeper for many analytics projects. The financial obligations for capturing, processing and managing analytics should be weighed against the value to the organization of the information, in terms of savings, in terms of identifying new areas of opportunity.

Prioritizing analytics should go beyond measures of profit, loss, cost, and expense. While financial information often appears accessible and transparent, numbers have a way of being manipulated to clever ends, so prioritizing what to measure must include other considerations. Engagement or satisfaction is relevant to learning activity, so in consideration of what to measure should not be limited simply to what informs investment and budget decisions. Consider how access to certain information about the learning experience may benefit the learner's experience directly and indirectly. How might information provide early warning signs of capacity challenges, knowledge gaps or overtraining?

Ultimately, what must be considered in prioritizing what to track is if the measurement is ethically sound. Protecting personal data and individual ethics concerns are top priority. How learning and performance success is interpreted will be reinforced and largely structured by the schema of an xAPI Profile.

Recognizing Likely Reasons for a new xAPI Profile

There are no technical rules related to the conditions under which an xAPI Profile is created. With seven years of informal and formal practices around learning data strategies and xAPI Profiles in particular, three scenarios in particular lend themselves to developing a new xAPI Profile.

1. **A wholly unique learning experience requires a data architecture.** *For example: a trade group is working on a competency model expressed with an xAPI Profile for normalized measurement and analysis of conformant learning activities so multiple publishers of training content, instructional aids, authoring tools and reporting services can work dynamically with the same set of semantically and technically interoperable frameworks.*
2. **A common definition for a modality of learning.** *For example: A standards group is charged with translating anticipated content/client data exchanges and user interactions into a measurable performance context. A hypothetical scenario could be: the Veterans Administration working with IEEE to standardize learning interactions for nurses within an Emergency Health Record System to evaluate against trackable patient outcomes.*
3. **An organization intends to govern its own xAPI data model.** *For example, the Naval Education and Training Center (NETC) codifies an xAPI Profile to normalize its capture and analysis of performance support activities, such as checklists, and their use for instructors and sailors. Adopters of the Total Learning Architecture (TLA) codifies an xAPI Profile describing valid statements and expected systemic outcomes that depend on particular properties of data captured.*

Authoring Guidance

Authoring an xAPI Profile in “Easy” Steps

In order to best prepare concepts in advance, a best practice is to begin with identifying the xAPI statements needed for the profile. One approach to determining that is thinking about which reports and other uses for the data will be needed, and architecting a collection of statements and patterns that would best support those outputs.

Once the needed statements are identified (A, for ALL the statements), further define the specifics of those statements by identifying the parts that need to be templated, what if any determining properties would help process data faster for reporting and what patterns align with reporting, analysis or other needs (B, for BATCH processing preparation). From this, define concepts unique to the profile that both support the statements needed AND are specifically designed/engineered to further inform the aims of the profile through their reuse in other xAPI Profiles, therefore reuse in other xAPI applications (C, for CONCEPTS).

The above A-B-C gets you to the 1-2-3 of authoring the xAPI Profile on the xAPI Profile Server.

1. First, add new and existing Concepts to the profile efficiently, correctly and confidently.
2. Second, define the profile Statement Templates and Patterns and any Determining Properties using the Concepts.
3. Publish!

Implementation Notes

A good xAPI Profile begins and ends with good documentation. Good implementation notes must provide complete and actionable answers with helpful contextual information to best impact successful adoption.

While the following list is by no means exhaustive, it is representative of the level of detail implementation guidance must provide to implement xAPI most efficiently.

1. What are the use cases driving the xAPI Profile?
2. What version of xAPI is supported?
3. What requirements need to be met for others to review and test, and to deliver the developed media?
4. What conventions must be followed?
5. What acronyms and terminology am I expected to know to read/follow the document?
6. What automated or human actions are triggered by specific data elements?
7. What patterns of xAPI statements are important to specific people or systems?
8. How and where in the xAPI statement should competencies be expressed?
9. What other specs, standards, published assertions and/or xAPI Profiles are referenced in a xAPI Profile?
10. What is the scope and purpose of the profile?
 - a. Is the profile going to define a small set of concepts used to define a domain (like the Open Badges profile)
 - b. Is the profile going to be an all-encompassing domain model used to inform xAPI instrumentation of a learning record provider?
 - c. How general or granular is the need being addressed?
 - d. Are you defining rules for a mode or archetype for media (e.g. an xAPI Profile for mobile applications or general simulations)?
 - e. Are you defining rules for a specific use case (e.g. an xAPI Profile for a specific professional accreditation program)?

An xAPI Profile may qualify as meeting a “Gold Standard” for xAPI Profiles if:

1. The xAPI Profile can sufficiently answer the above questions, such that...
2. The answers to the above questions can fully inform a complete set of documentation to construct a JSON-LD xAPI Profile document that expresses its intent, such that...
3. At least three independent development efforts can produce components that generate xAPI data that can demonstrate
 - a. Conformance to the xAPI Profile
 - b. Semantically interoperable data generation and/or consumption

Statement Templates

Statement templates provide a set of instructions for what data is required in an xAPI statement which describes a particular event. The group creating a profile should be writing corresponding documentation describing instructional design considerations and implementation decisions for each statement template. This is the documentation for the profile, beyond the json-ld stored in the profile server. The server links to this documentation at the profile level to describe the intent of each part of the profile and the intended use. In preparing a statement template, the first activity should be to research what is already defined as a concept on the xAPI Profile Server.

Consider the aforementioned reasons for a new xAPI Profile with the following example.

In this example, let's assume there is a need for an xAPI statement that expresses when a learner meets all the requirements to earn a software certification, but has not yet been conferred or awarded the certification. To look for a statement template that may meet the needs (see: [*xAPI Profile Server User Guide > Statement Templates > Add existing statement template*](#)), one could search the server for a "Qualified" statement. Such a search could yield a result for a "Qualified" verb. As defined in the Master Object Model xAPI Profile, the .json-ld may look like this:

```
{
  "id": "https://w3id.org/xapi/tla#qualified",
  "type": "StatementTemplate",
  "prefLabel": {
    "en": "qualified"
  },
  "definition": {
    "en": "Indicates the learner meets all the requirements for a badge, but hasn't been awarded the badge yet"
  },
  "inScheme": "https://w3id.org/xapi/tla/v0.10",
  "verb": "https://w3id.org/xapi/tla/verbs/qualified",
  "rules": [
    {
      "location": "$.result",
      "presence": "excluded"
    },
    {
      "location": "$.object.id",
      "presence": "included",
      "scopeNote": {
        "en": "This should point to a badge"
      }
    },
    {
      "location": "$.object.definition.type",
      "presence": "included",
      "any": [
        "https://w3id.org/xapi/tla/activity-types/badge"
      ]
    }
  ]
}
```

```

    },
    {
      "location":
        "$.context.extensions['https://w3id.org/xapi/tla/extensions/evidence']",
      "presence": "included",
      "scopeNote": {
        "en": "This should be a pointer to the xAPI statement(s) that resulted in
              this statement"
      }
    }
  ]
},

```

If this statement template could be used as-is in this example profile, it could be included as-is. The rules of the statement template in the above example can be understood as “A ‘Qualified’ xAPI statement is valid to this xAPI Profile if...”

- It includes the determining property of “qualified” as the required verb in the statement
- It doesn’t have a Result object
- It includes an `object.id` that should point to a badge
- It includes an `object.definition.type` of “badge”
- It includes a link to the “evidence” as a extension in the statement

Adopting this statement template as-is, the statements could never include a result object when statements use the ‘qualified’ verb and include the example profile ID. If there was a need for a Result object, the above statement template would not be appropriate and should not be used as-is from the source profile.. This means a new statement template should be created instead of reusing a statement template above, which is owned by another profile (see: [xAPI Profile Server User Guide > Statement Templates > Add rules](#)).

Note: When similar statement templates exist, but they are not suitable for use as they are written, they may be linked to the profile’s statement template as a statement template reference. This allows people implementing the statement template a way to understand the similarities. This also helps with linking like statement templates together (see: [xAPI Profile Server User Guide > Statement Templates > Create new statement template > Statement Template References](#)).

Determining Properties

Determining properties are used to identify which statements will be validated, to which statement templates. This mapping is required because xAPI statements do not include statement template IDs. All selected determining properties must be included in an xAPI statement to be validated to the statement template.

Only one identifier may be selected for the following determining properties:

- verb
- objectActivityType

Multiple identifiers may be selected for the following determining properties:

- contextCategoryActivityType
- contextGroupingActivityType
- contextOtherActivityType
- contextParentActivityType
- attachmentUsageType

A rule describes the data requirements for a statement to be conformant to the statement template. Rules have a huge amount of flexibility. In order to allow this flexibility to be used as the specification designed, rules must be written as json. Fortunately, the xAPI Profile Server has a wizard to produce the needed json. See [*xAPI Profile Server User Guide > Statement Templates > Add rules*](#).

Rules

There are two ways to set rules for a statement. One is through presence and the other through specifying values. *Presence* describes which locations in an xAPI statement are important to the statement. If the goal is to receive data in a specific location, but it is not important what data is to be in the location, "presence": "included" is the best way to do that.

Presence

Presence describes whether a specified location should be used to conform to this template.

- **included**, the location must be present with a value in every statement to conform to this template
- **excluded**, this location should not be in any statement to conform to this template
- **recommended**, this location should be present, but it will not fail validation to this template if it is excluded

Example of Presence:

```
"rules": [  
  {  
    "location": "context.contextActivities.grouping[*].id",  
    "presence": "included"  
  }  
]
```

In this example, a statement must include an ID for every activity in the context activities grouping array in order to conform to this statement template.

Specifying Values

If just saying whether or not a location should be in the statement is not enough, the following may be used to identify which data will allow a statement to validate to this template.

- **any**, any of the values listed are allowed to be in the specified location in order for a statement to validate to a template. When any is used, a list of potential values must be included. A statement will validate to this template if at least one of the values listed is included in the specified location in the template.
- **all**, all of the values in the specified location must be listed in the all array in order for a statement to validate to a template. When all is used, a list of values must be included. A statement will validate if ALL of those values are present in the statement.
- **none**, none of the values listed are allowed to be in the specified location in order for a statement to validate to a template. When none is used, a list of potential values must be included. A statement will validate to this template if none of the values listed is included in the specified location in the template.

Example of using any:

```
"rules": [
  {
    "location": "context.contextActivities.grouping[*].definition.type",
    "any": ["http://adlnet.gov/expapi/activities/attempt"]
  }
]
```

In this example, a statement can use the activity type of "http://adlnet.gov/expapi/activities/attempt" as the definition of type for context activities grouping as at least one of the provided definition types and the statement will conform to this statement template.

Once a rule is written into the xAPI Profile Server, the rule will be validated before it can be added to the statement template.







The following validation errors may present themselves:

- Improper json
- Unknown identifier
- Conflict with determining properties
- Value not allowed

Once a rule is successfully validated, it may be added to the statement template. As concepts are defined for use in determining properties and rules, they are added to the list of associated concepts. These associated concepts are intended to provide an overview of the vocabulary used in the statement template.

Concepts

Concepts are building blocks for use and reuse across xAPI Profiles. In the case of Verbs, Activity Types, Attachment Usage Types, and Activities, the Concept is "the thing" you use within an xAPI Profile. In the case of Document Resources and Extensions, the Concept is essentially a variable or container to hold many different values of xAPI data. Concepts should be defined within an xAPI Profile. When a concept is defined by a profile, that profile is the owner of the concept and responsible for the concept's management (see: [xAPI Profile Server User Guide > Concepts](#)).

					
Activity	Activity Type	Attachment Usage Type	Document	Extension	Verb
Identifies a specific interaction with an object by an actor in a statement. It can be a unit of instruction, experience, or performance that is to be tracked in meaningful combination with a Verb.	Identifies the type of activity in a broader category. For example, a course, video, book, or assessment.	Identifies the type of usage of this Attachment. For example, one expected use case for Attachments is to include a "completion certificate".	Identifies information about the data to be stored in the State, Agent Profile, and Activity Profile Resources.	Identifies information about the data to be stored in the Context, Result, and Activity Extensions.	Identifies the action to be done by the actor a statement.

A goal for xAPI Profiles is to avoid duplications of definitions. Reusing concepts from other profiles is highly recommended when appropriate. If a concept is not already defined on the profile server, that is when it should be defined within a new profile.

01

Patterns and templates provide the starting set of needed concepts to be included in the profile, from the top-down.

02

With those, from the bottom-up, identify (but don't define just yet) concepts as the building blocks for statements.

03

Compare that list to previously identified concepts in the xAPI Profile Server.

04

Research existing profiles to avoid duplication of existing definitions.

In the aforementioned Statement Template example, the activity-type is <https://w3id.org/xapi/tla/activity-types/badge>. Activity-types must be defined in the xAPI Profile Server before they can be referenced in defining activities. For example, one could not define a specific badge that a learner could earn without first defining the Concept of the badge activity-type.

Regardless of the reason for creating a new xAPI Profile, to look for potential concepts already defined that may meet the needs (see: *xAPI Profile Server User Guide > Concepts > Reusing Concepts from Other Profiles*), one could search the server for a “badge”

Patterns

Patterns describe requirements for grouping or substitution of other Patterns or Statement Templates. For example, a Pattern in a video Profile might start with a Statement about playing a video and then be followed by Statements about pausing, skipping, playing again, and so forth.

When adding patterns, much like Statement Templates and Concepts, the first step should be to reuse existing patterns from other profiles. One or more patterns may be selected via search and added to the profile. If no existing patterns match the profile's needs, a new pattern may be created.

The first step in creating a new pattern is selecting the type of pattern. When considering the pattern type, it is important to know if this will be a primary or secondary pattern. Designating the pattern primary or secondary will be done on the next screen. The only time a pattern may be selected as a component of another pattern is when it is a secondary pattern.

The screenshot shows the 'Add Pattern' interface. At the top, there's a search bar with the text 'keyword' and a 'Search' button. Below the search bar, it says '4 results for "keyword"'. There are four results listed, each with a 'Pattern Name', a description, and a 'View info' link. The first two results are marked as 'Selected' with a blue button. The third result is marked as 'Select' with a blue button. The fourth result is marked as 'Select' with a blue button. On the right side, there's a 'Selected (2)' section showing two selected patterns, each with a 'Remove' button and a 'View info' link. At the bottom, there are three buttons: 'Cancel', '+ Create New', and 'Add Selected to Profile'.

Primary

Primary patterns define which combinations of statements have meaning to a profile. A primary pattern cannot be pulled into other patterns. Each statement received with this profile ID will be checked for conformance to this pattern.

Secondary

Contains statement templates and secondary patterns. Statements received which contain this profile ID will not be checked for conformance to this pattern, unless it is pulled into the primary pattern for this profile.

The following pattern types are available:

- **Sequence** - A list of statement templates or secondary patterns in a particular order. This will match if the identified components match in the order specified. This type allows selection of multiple components and ordering of them in a three-step process.
- **Alternates** - A list of statement templates or secondary patterns. This will match if any component from the list matches. This type allows selection of multiple components, but not ordering because alternates are interchangeable with one another.

- **Optional** - Identifies a single statement template or secondary pattern. This will match if the component is not present once or not at all. This type allows selection of a single component which is not required, but may be used once, for a pattern to be considered complete.
- **One or More** - Identifies a single statement template or secondary pattern. This will match if the component is present one or more times. This type allows selection of a single component which must be used once and may be used more for a pattern to be considered complete.
- **Zero or More** - Identifies a single statement template or secondary pattern. This will match if the component is not present or is present one or more times. This type allows selection of a single component which is not required, but may be used multiple times, for a pattern to be considered complete.

Patterns will likely be used to describe learning event “triggers” that would happen in domain-specific architectures, like the earning of a badge, or successfully meeting the requirements of a recertification. If an xAPI Profile is going to use patterns in this way, then one should employ, with care, how statement templates are defined as well as how pattern types will be utilized.



Strictly controlling the series of statements required to trigger a response reduces the ability of any other statements to match a pattern thus causing an incorrect response to a series of statements.

Example: In the Video Profile, the matches against the `all-activities-pattern` any of the defined statement templates, whereas the `general-pattern` specifies that the first statement in a sequence of statements must match against the started template. The three defined patterns within the Video Profile are sufficient to match against any sequence of interactions a person can have with a video.